

AN ALGORITHM FOR THE
QUADRATIC APPROXIMATION

by

A. W. McInnes

and

S. L. Loi *

No. 34

June, 1984

Abstract

The quadratic approximation is a three dimensional analogue of the two dimensional Pade approximation. A determinantal expression for the polynomial coefficients of the quadratic approximation is given. A recursive algorithm for the construction of these coefficients is derived. The algorithm constructs a table of quadratic approximations analogous to the Pade table of rational approximations.

* Department of Mathematics,
University of Singapore.

1. INTRODUCTION

Pade approximations are widely used as a form of approximation more general than polynomial approximations. Recently there has been some interest in generalizations and extensions of Pade approximations [4,11,12]. The Pade approximation has been extended to quadratic approximation [9,13] and used in applications in the physical sciences to approximate certain functions with branch points [6,14]. Some of these results show that the quadratic approximation provides a better approximation than the Pade approximation. Essentially the approximation is primarily used as a practical method of analytic continuation. For appropriate functions, the quadratic approximation represents the function over a wider range of the independent variable than either the power series or Pade representations.

The aim of this paper is to present a recursive algorithm for the construction of quadratic approximations. A general form for the expression of the quadratic approximations in terms of the polynomial coefficients is given in the next section. Identities among these polynomial coefficients are given in section 3 and a recursive algorithm for computing these polynomial coefficients is given in section 4. This algorithm is an extension of the general extrapolation algorithm [3] and the MNA algorithm [10,2]. It also extends a similar algorithm for Pade approximations [8]. Two examples of the use of the algorithm are given in section 5.

2. THE QUADRATIC APPROXIMATION

It is convenient to begin by reviewing the usual Pade approximation.

The Pade approximation $R(m,n;x)$ satisfies the equation

$$q(m,n;x) R(m,n;x) + r(m,n;x) = 0 \quad (2.1)$$

where the numerator $r(m,n;x)$ is a polynomial of degree n and the denominator $q(m,n;x)$ is a polynomial of degree m . The polynomial coefficients q, r are determined by the 'accuracy-through-order' conditions

$$q(m,n;x)f(x) + r(m,n;x) = O(x^{m+n+1})$$

where $R(m,n;x)$ approximates the given function $f(x)$ which can be expressed as a formal power series $\sum_{j=0}^{\infty} f_j x^j$. This representation is usually assumed to satisfy the normalization condition

$$q(m,n;0) = 1.$$

Note that this notation is that of the classical definitions. The degrees of the numerator and denominator are the reverse of the notation introduced by Baker [1] and subsequently used by many authors.

The polynomial coefficients can be expressed as the quotient of a determinant of order $m+1$ and its principal minor (compare [1]). Let $a(m,n;x)$ represent q and r . Then

$$a(m,n;x) = D^{-1} \cdot \begin{vmatrix} F_n(x) & xF_{n-1}(x) & \dots & x^m F_{n-m}(x) \\ f_{n+1} & f_n & \dots & f_{n-m+1} \\ \dots & \dots & \dots & \dots \\ f_{n+m} & f_{n+m-1} & \dots & f_n \end{vmatrix} \quad (2.2)$$

where (i) D is the minor of $F_n(x)$ in the numerator

(ii) $f_k = 0$ if $k < 0$

(iii) For $a(m,n;x) = q(m,n;x)$ set $F_k(x) = 1$, $\forall k$

For $a(m,n;x) = r(m,n;x)$ set $F_k(x) = \begin{cases} -\sum_{j=0}^k f_j x^j, & k \geq 0 \\ 0, & k < 0 \end{cases}$

Note that $q(m,n;0) = D^{-1} \cdot D = 1$ as required.

An algorithm for the computation of these coefficients is given in [8].

The quadratic approximation $Q(\ell, m, n; x)$ satisfies the equation

$$p(\ell, m, n; x)Q^2(\ell, m, n; x) + q(\ell, m, n; x)Q(\ell, m, n; x) + r(\ell, m, n; x) = 0, \quad (2.3)$$

where p, q, r are polynomial of degree ℓ, m, n respectively. If $Q(\ell, m, n; x)$ is to approximate the function $f(x) = \sum_{j=0}^{\infty} f_j x^j$ we impose the 'accuracy-through-order' conditions

$$p(\ell, m, n; x)f^2(x) + q(\ell, m, n; x)f(x) + r(\ell, m, n; x) = 0(x^{\ell+m+n+2})$$

in order to determine the polynomial coefficients p, q, r . In this representation we assume the normalization condition

$$p(\ell, m, n; 0) = 1.$$

The polynomial coefficients can be expressed as the quotient of a determinant of order $\ell+m+2$ and its principal minor. Let $a(\ell, m, n; x)$ represent p, q and r and let

$$g(x) = f^2(x) = \left(\sum_{j=0}^{\infty} f_j x^j\right)^2 = \sum_{j=0}^{\infty} g_j x^j.$$

Then

$$a(\ell, m, n; x) = D^{-1}.$$

$$\begin{vmatrix} G_n(x) & xG_{n-1}(x) & \dots & x^{\ell}G_{n-\ell}(x) & F_n(x) & xF_{n-1}(x) & \dots & x^mF_{n-m}(x) \\ g_{n+1} & g_n & \dots & g_{n-\ell+1} & f_{n+1} & f_n & \dots & f_{n-m+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_{N+1} & g_N & \dots & g_{m+n+1} & f_{N+1} & f_N & \dots & f_{\ell+n+1} \end{vmatrix} \quad (2.4)$$

where (i) $N = \ell+m+n$

(ii) D is the minor of $G_n(x)$ in the numerator

(iii) $f_k = 0, g_k = 0$ if $k < 0$.

(iv) For $a(\ell, m, n; x) = p(\ell, m, n; x)$ set $F_k(x) = 0, G_k(x) = 1, \forall k$

For $a(\ell, m, n; x) = q(\ell, m, n; x)$ set $F_k(x) = 1, G_k(x) = 0, \forall k$

For $a(\ell, m, n; x) = r(\ell, m, n; x)$ set $F_k(x) = \begin{cases} -\sum_{j=0}^k f_j x^j, & k \geq 0 \\ 0, & k < 0 \end{cases}$

$$G_k(x) = \begin{cases} -\sum_{j=0}^k g_j x^j, & k \geq 0 \\ 0, & k < 0 \end{cases}$$

The quadratic approximation, $Q(\ell, m, n; x)$, to $f(x)$ is given implicitly by the equation (2.3), where the polynomial coefficients satisfy the relations (2.4). An explicit form of the approximation is obtained as a root of the quadratic equation (2.3). The selection of the root requires some comment. When seeking the zero of a function, the root closest to zero is selected (Cauchy's Method). Thus the selected root may be written as [13]

$$(-q + \sqrt{q^2 - 4pr})/2p = -2r/(q + \sqrt{q^2 - 4pr})$$

In the case of extrapolation the root closest to the previous approximation is selected. This is the procedure used for the numerical example in section 5. These results show that the approximant may shift between the two roots of the quadratic equation.

3. IDENTITIES AMONG THE POLYNOMIAL COEFFICIENTS OF THE QUADRATIC APPROXIMATION

Since the three polynomial coefficients have common denominators, in this section we may 'rationalize' the coefficients in equation (2.3) by multiplication by the denominator. We derive relationships among the numerators in the expressions (2.4).

Let $a(m,n;x)$ denote, in turn, the polynomial coefficients $q(m,n;x), r(m,n;x)$ of the Pade approximation of $f(x) = \sum_{j=0}^{\infty} f_j x^j$. With the appropriate change of notation, we obtain from

[1,p33] the following identity between the polynomial coefficients of the Pade approximation:

$$a(m,n;x) = [a(m-1,n-1;x) \cdot a(m-1,n+1;x) - a(m-1,n;x)^2] / a(m-2,n;x)$$

for $m = 1, 2, \dots; n = 0, 1, 2, \dots$ (3.1)

The initialization is:

$$\begin{aligned} \text{(i)} \quad \text{If } a = r \text{ then } r(m,n;x) &= x^n && \text{for } m = -1, n \geq 0; \\ &= -\sum_{j=0}^n f_j x^j && \text{for } m = 0, n \geq 0; \\ &\quad \text{(Taylor polynomial)} \\ &= 0 && \text{for } n = -1, m \geq 0; \\ \text{(ii)} \quad \text{If } a = q \text{ then } q(m,n;x) &= 1 && \text{for } m = 0, n \geq 0; \\ &= f_n - x f_{n+1} && \text{for } m = 1, n \geq 0; \\ &= (-f_0 x)^m && \text{for } n = -1, m \geq 0; \end{aligned}$$

Similarly, let $\bar{q}(\ell,n;x), \bar{r}(\ell,n;x)$ denote the polynomial coefficients of the Pade approximation of $g(x) = f^2(x)$. Then these coefficients must also satisfy the identity (3.1).

The numerator of $a(\ell,m,n;x)$ in (2.4) is decomposed using Sylvester's determinant identity. The following relations are obtained by using identity (3.1):

Except for $p(0, m, n; x)$ and $q(\ell, 0, n; x)$
 for $\ell = 1, 2, \dots, m, n = 0, 1, 2, \dots$,
 or $m = 1, 2, \dots, n = 0, 1, 2, \dots$,
 we have

$$a(\ell, m, n; x) = \frac{[a(\ell-1, m, n; x) \cdot a(\ell, m-1, n+1; x) - a(\ell, m-1, n; x) \cdot a(\ell-1, m, n+1; x)]}{a(\ell-1, m-1, n+1; x)} \quad (3.2)$$

The initialization is:

- (i) $a(\ell, m, n; x) = G_n(x) f_{n+1} - F_n(x) g_{n+1}$ for $\ell = m = 0$
- (ii) If $a = r$ then $r(\ell, m, n; x) = r(m, n; x)$ for $\ell = -1$
 $= \bar{r}(\ell, n; x)$ for $m = -1$
- (iii) If $a = q$ then $q(\ell, m, n; x) = q(m, n; x)$ for $\ell = -1$
- (iv) If $a = p$ then $p(\ell, m, n; x) = \bar{q}(\ell, n; x)$ for $m = -1$

Note that if the exceptional cases $p(0, m, n; x)q(\ell, 0, n; x)$ were to be defined by the relation (3.2) we would need the initial values $p(-1, m, n; x), q(\ell, -1, n; x)$ respectively, but these initial values are undefined in the initialization above. These exceptional cases may be defined by

$$p(0, m, n; x) = \frac{[p(0, m-1, n-1; x) \cdot p(0, m-1, n+1; x) - p(0, m-1, n; x)^2]}{p(0, m-2, n; x)}$$

$$q(\ell, 0, n; x) = \frac{[q(\ell-1, 0, n-1; x) \cdot q(\ell-1, 0, n+1; x) - q(\ell-1, 0, n; x)^2]}{q(\ell-2, 0, n; x)}$$

where $p(0, m, -1; x) = f_0^{m+1}$, $q(\ell, 0, -1; x) = g_0^{\ell+1}$

Since $p(0, m, n; x)$ is a constant polynomial, the assumed normalization of the polynomial coefficients mentioned in the previous section may be achieved by division of (2.3) by $p(0, m, n; x)$ (where the coefficients are in the 'rationalized' form of this section).

The relationship between the $a(\ell, m, n; x)$ of different orders is shown in Figure 3.1

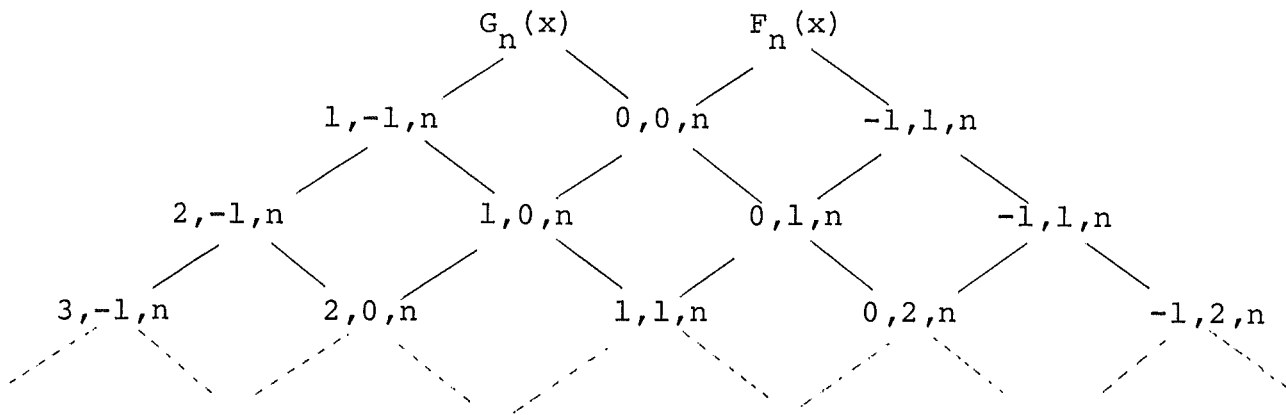


Figure 3.1

4. THE GENERAL ALGORITHM FOR THE COMPUTATION OF THE POLYNOMIAL COEFFICIENTS OF THE QUADRATIC APPROXIMATION

The polynomial coefficients of equation (2.3) can be expressed by the relations (2.4). These polynomial coefficients can be computed recursively by the relations (3.1), (3.2). However since these latter relations involve products, the evaluation of the polynomial coefficients often caused overflow or underflow during the computation.

A general recursive extrapolation algorithm has recently been found by Brezinski [3]. This is closely related to the general interpolation algorithm [10] called the MNA algorithm [2]. The MNA algorithm has been extended Pade approximation [8] and rational interpolation [7]. By the analogue of this algorithm, the polynomial coefficients of the quadratic approximation of $f(x)$ may be computed recursively and

effectively in a more general way. Assuming $f(x)$ has a formal power series $\sum_{j=0}^{\infty} f_j x^j$, each polynomial coefficient of the quadratic approximation to $f(x)$ can be computed in turn by the following algorithm.

4.1 Basic recursion formulas:

We note from Figure 3.1 that the coefficients $a(\ell, m, n; x)$ may be computed either from $a(\ell-1, m, n; x)$ or from $a(\ell, m-1, n; x)$.

To implement the algorithm we need to introduce some auxiliary functions. For each coefficient a (i.e. p, q or r) we need two auxiliary functions ag_1, ag_2 , depending on whether we choose to increment the first or second argument to obtain $a(\ell, m, n; x)$. These auxiliary functions are generated by the recursions:

$$ag_1(\ell, m, n; x) = ag_1(\ell, m-1, n; x) - \frac{\Delta ag_1(\ell, m-1, n; x)}{\Delta ag_2(\ell-1, m, n; x)} ag_1(\ell-1, m, n; x) \quad (4.1)$$

$$ag_2(\ell, m, n; x) = ag_2(\ell-1, m, n; x) - \frac{\Delta ag_2(\ell-1, m, n; x)}{\Delta ag_1(\ell, m-1, n; x)} ag_2(\ell-1, m, n; x) \quad (4.2)$$

where Δ represents the forward difference on the index n .

With these auxiliary functions we may define the alternative recursions for the polynomial coefficients $a(\ell, m, n; x)$:

$$a(\ell, m, n; x) = a(\ell-1, m, n; x) - \frac{\Delta a(\ell-1, m, n; x)}{\Delta ag_1(\ell, m, n; x)} ag_1(\ell, m, n; x) \quad (4.3)$$

$$a(\ell, m, n; x) = a(\ell, m-1, n; x) - \frac{\Delta a(\ell, m-1, n; x)}{\Delta ag_2(\ell, m, n; x)} ag_2(\ell, m, n; x) \quad (4.4)$$

where Δ represents the forward difference on the index n .

4.2 Theoretical justification:

THEOREM 1 (A) For all integers $\ell \geq 2$, $m \geq 1$, $n \geq 0$

$$(i) \quad a(\ell, m, n; x) = a(\ell, m-1, n; x) - \frac{\Delta a(\ell, m-1, n; x)}{\Delta a g_2(\ell, m, n; x)} a g_2(\ell, m, n; x)$$

$$(ii) \quad a(\ell, m, n; x) = a(\ell-1, m, n; x) - \frac{\Delta a(\ell-1, m, n; x)}{\Delta a g_1(\ell, m, n; x)} a g_1(\ell, m, n; x)$$

(B) In addition (i) holds for all $a(\ell, m, n; x)$ if

$\ell = 1$, $m \geq 1$, $n \geq 0$, and holds for $a = q, r$ if $\ell = 0$, $m \geq 1$, $n \geq 0$.

Further, (ii) holds for all $a(\ell, m, n; x)$ if $\ell \geq 2$, $m = 0$, $n \geq 0$,

and holds for $a = q, r$ if $\ell = 1$, $m \geq 0$, $n \geq 0$.

Proof:

(A) (i) This can be proved by using the method in [2].

The proof here summarizes the more detailed proof used previously [9]. Using the Sylvester identity for determinants, $a(\ell, m, n; x)$ in (2.4) can be decomposed in the form:

$$a(\ell, m, n; x) = a(\ell, m-1, n; x) -$$

$$\frac{\begin{vmatrix} xG_{n-1}(x) \dots x^{\ell}G_{n-\ell}(x) & F_n(x) \dots x^mF_{n-m}(x) \\ g_n \cdot \dots \cdot g_{n-\ell+1} & f_{n+1} \cdot \dots \cdot f_{n-m+1} \\ \vdots & \vdots \\ g_{N-1} \cdot \dots \cdot g_{m+n} & f_N \cdot \dots \cdot f_{\ell+n} \end{vmatrix}}{\begin{vmatrix} g_{n+1} \cdot \dots \cdot g_{n-\ell+1} & g_{n+1} \cdot \dots \cdot f_{n-m+2} \\ \cdot & \cdot \\ \vdots & \vdots \\ g_{N+1} \cdot \dots \cdot g_{m+n+1} & f_{N+1} \cdot \dots \cdot f_{\ell+n+2} \end{vmatrix}},$$

$$\frac{\begin{vmatrix} g_n \cdot \dots \cdot g_{n-\ell+1} & f_{n+1} \cdot \dots \cdot f_{n-m+2} \\ \cdot & \cdot \\ g_{N-1} \cdot \dots \cdot g_{m+n} & f_N \cdot \dots \cdot f_{\ell+n+1} \end{vmatrix}}{\begin{vmatrix} g_n \cdot \dots \cdot g_{n-\ell+1} & f_{n+1} \cdot \dots \cdot f_{n-m+1} \\ \cdot & \cdot \\ g_N \cdot \dots \cdot g_{m+n+1} & f_{N+1} \cdot \dots \cdot f_{\ell+n+1} \end{vmatrix}}$$

$$= a(\ell, m-1, n; x) - a g_2(\ell, m, n; x) \cdot \frac{\Delta a(\ell, m-1, n; x)}{\Delta a g_2(\ell, m, n; x)}$$

We define the first ratio of determinants in the second term of the right hand side as the auxiliary function $ag_2(\ell, m, n; x)$. The second ratio can be shown [9] to be equal to the ratio of the forward differences on n of $a(\ell, m-1, n; x)$ and $ag_2(\ell, m, n; x)$ by using the same identity and some elementary operations on the determinants.

(A) (ii) This result is shown by the same method as in (A) (i), using a slightly modified form of Sylvester's identity. (Alternatively shift the $(\ell+1)$ st column of (2.4) to the last column and use the standard form of the Sylvester identity.) In the case the first ratio of determinants in the second term is defined to be $ag(\ell, m, n; x)$.

(B) (i) If $\ell = 1$ then the proof of (A) (i) still holds. If $\ell = 0$ the proof breaks down in the case $a = p$, since $pg_2(0, m, n; x) = 0$ for all n . However the imposed normalization condition implies that $p(0, m, n; x) = 1$ and hence these values do not need to be calculated.

(B) (ii) If $m = 0$ then the proof of (A) (ii) still holds. If $\ell = 1$ the proof breaks down in the case $a = p$, since $pg_1(1, m, n; x) = x$ for $m \geq 0$, $n \geq 0$, and hence $\Delta pg_1(1, m, n; x) = 0$ $\Delta pg_1(1, m, n; x) = 0$. However as noted above in (B) (i), $p(0, m, n; x) = 1$ and hence $\Delta p(0, m, n; x) = 0$. Thus the ratio $\Delta p(0, m, n; x) / \Delta pg_1(1, m, n; x)$ is not defined. It is clear from (A) (ii) that this ratio does have a definite value and the decomposition is still true, but its expression in terms of differences for the computational algorithm cannot be used.

Notes

1. The expressions for $ag_1(\ell, m, n; x)$ and $ag_2(\ell, m, n; x)$ have common numerators and the denominators have one column different.
2. The expressions for $ag_1(\ell, m-1, n; x)$ and $ag_2(\ell-1, m, n; x)$ have common denominators and the numerators have one column different.

COROLLARY 1

(A) For all integers $\ell, m, n \geq 0$ and $\ell+m \neq 1$

$$(i) \quad ag_1(\ell, m, n; x) = ag_1(\ell, m-1, n; x) - \frac{\Delta ag_1(\ell, m-1, n; x)}{\Delta ag_2(\ell-1, m, n; x)} \cdot ag_2(\ell-1, m, n; x)$$

$$(ii) \quad ag_2(\ell, m, n; x) = ag_2(\ell-1, m, n; x) - \frac{\Delta ag_2(\ell-1, m, n; x)}{\Delta ag_1(\ell, m-1, n; x)} \cdot ag_1(\ell, m-1, n; x)$$

(B) For $\ell > 2$, $m < 0$ and $\ell+m \neq 1$, $n \geq |m|$ (i) and (ii) hold

(C) For $m > 2$, $\ell < 0$ and $\ell+m \neq 1$, $n \geq |\ell|$ (i) and (ii) hold

Proof: The proof of the recursions for the auxiliary functions is similar to that for Theorem 1.

COROLLARY 2

If the ratios $\frac{\Delta a(\ell-1, m, n; x)}{\Delta ag_1(\ell, m, n; x)}$, $\frac{\Delta a(\ell, m-1, n; x)}{\Delta ag_2(\ell, m, n; x)}$

$$\frac{\Delta ag_1(\ell, m-1, n; x)}{\Delta ag_2(\ell-1, m, n; x)} , \frac{\Delta ag_2(\ell-1, m, n; x)}{\Delta ag_1(\ell, m-1, n; x)}$$

are defined, they are constants depending on ℓ, m, n (but independent of x) and each ratio has the same value for $a = p, q, r$.

Proof: The second ratio is clearly a constant in equation (4.5). The other ratios follow similarly.

COROLLARY 3

$$ag_1(\ell, m, n; x) = -K ag_2(\ell, m, n; x) \quad \text{where}$$

$$K = \Delta ag_1(\ell, m-1, n; x) / \Delta ag_2(\ell-1, m, n; x)$$

is a constant depending only on ℓ, m, n .

Proof: This follows immediately by combining (4.1) and (4.2) since the ratios of the differences are inverses and, by Corollary 2, constant.

Since this algorithm is an extension of the algorithm obtained for generalized rational interpolation [7], this theorem and corollaries are similar to the theorem and corollaries obtained in [7].

By Corollary 2, the ratios in the basic recursion relations (4.3) and (4.4) are constants independent of x and independent of the particular coefficient represented by a . Although it is a convenient formalism to express the ratio in terms of the corresponding polynomial coefficient (i.e. p, q or r), it was shown in the proof of Theorem 1 (B) that this formalism may become undefined. However by taking advantage of the independence of the ratio with respect to a particular polynomial coefficient it is computationally convenient to express all the ratios in terms of $a = r$ in the algorithm to avoid a breakdown.

For a compact expression of the algorithm it is convenient to formally define the auxiliary functions for all negative argument values of either ℓ or m [Note Corollary 1 (B), (C)]. To compute $ag_1(\ell, -j, n; x)$ and $ag_1(-j, m, n; x)$ for

increasing values of n , n begins at the value j . This follows since for $n < j$, $ag_2(\ell, -j, n; x)$ and $ag_1(-j, m, n; x)$ do not exist. Also for $n < j-1$, $ag_1(\ell, -j, n; x)$ and $ag_2(-j, m, n; x)$ do not exist. Although $ag_1(\ell, -j, j-1; x)$ and $ag_2(-j, m, j-1; x)$ cannot be computed by the algorithm [since $ag_2(\ell-1, -j, j-1; x)$, $ag_1(\ell, -j-1, j-1; x)$, $ag_1(-j, m-1, j-1; x)$ and $ag_2(-j-1, m, j-1; x)$ do not exist] the previously defined expressions for $ag_1(\ell, -j, n; x)$ (for $\ell > j$) and $ag_2(-j, m, n; x)$ (for $m > j$) reduce to zero in the case $n = j-1$. Hence they can be defined (or initialized) in the algorithm as

$$ag_1(\ell, -j, j-1; x) = 0$$

and

$$ag_2(-j, m, j-1; x) = 0$$

for

$$j = 1, 2, 3, \dots$$

4.3 The Algorithm:

This algorithm calculates a systematic table of quadratic approximations. Thus for a given order, N , the polynomial coefficients of the quadratic approximation, Q , are calculated for every combination of ℓ, m, n such that $\ell + m + n = N$.

The power series coefficients f_0, f_1, \dots, f_{N+1} of $f(x)$ and g_0, g_1, \dots, g_{N+1} of $g(x) = f^2(x)$ are assumed to be given.

Step 1. Set $NT = N + 2$

For $L = 1; n = 0, 1, \dots, N$

$$\text{Initialize: } ag_1(1, 0, n; x) = xG_{n-1}(x) - g_n F_n(x) / f_{n+1}$$

$$ag_2(0, 1, n; x) = xF_{n-1}(x) - f_n F_n(x) / f_{n+1}$$

$$a(0, 0, n; x) = G_n(x) - g_{n+1} F_n(x) / f_{n+1}$$

Step 2. For $n = 0, 1, \dots, N-1$

Compute: $a(1, 0, n; x)$ by (4.3)

$a(0, 1, n; x)$ by (4.4)

Step 3. Terminating criterion.

If $L = N_T - 2$ stop

Step 4. Set $N = N_T$

For $n = L-1, L, L+1, \dots, N-1$

Initialize:

$$ag_2(L, 1-L, n; x) = x^{L-1} G_{n-L+1}(x) - g_{n-L+2} x^L G_{n-L}(x) / g_{n-L+1}$$

$$ag_1(1-L, L, n; x) = x^{L-1} F_{n-L+1}(x) - f_{n-L+2} x^L F_{n-L}(x) / f_{n-L+1}$$

Set $L = L+1$

$$ag_1(L, 1-L, L-2; x) = 0$$

$$ag_2(1-L, L, L-2; x) = 0$$

For $n = L-1, L, \dots, N-1$

Initialize:

$$ag_1(L, 1-L, n; x) = x^L G_{n-L}(x) - g_{n-L+1} x^{L-1} G_{n-L+1}(x) / g_{n-L+2}$$

$$ag_2(1-L, L, n; x) = x^L F_{n-L}(x) - f_{n-L+1} x^{L-1} F_{n-L+1}(x) / f_{n-L+2}$$

Step 5. Set $N = N - 1$

(A) For $j = L-2, L-3, \dots, 0$

If $y = 0$ go to B.

$$\text{Set: } ag_1(L, -j, j-1; x) = 0$$

$$ag_2(-j, L, j-1; x) = 0$$

(B) For $n = j, j+1, \dots, N-1$

Compute: $ag_1(L, -j, n; x)$ by (4.1)

$ag_2(-j, L, n; x)$ by (4.2)

If $j = 0$ go to step 6

For $n = j-1, j, \dots, N-1$

Compute: $ag_2(L-1, -j+1, n; x)$ by (4.2)

$ag_1(-j+1, L-1, n; x)$ by (4.1)

Step 6. For $j = L-1, L-2, \dots, 1$

For $n = 0, 1, \dots, N-1$

For $i = 1, 2$

Compute: $ag_i(j, L-j, n; x)$ by (4.1) and (4.2)

Step 7. For $n = 0, 1, \dots, N-2$

Compute: $a(L, 0, n; x)$ by (4.3)

$a(0, L, n; x)$ by (4.4)

For $j = L-1, L-2, \dots, 1$

For $n = 0, 1, \dots, N-2$

Compute: $a(j, L-j, n; x)$ by (4.3) or (4.4)

Go to step 3.

In this algorithm the coefficients are computed a "level" at a time, where the level is defined by $L = \ell + m$, where $\ell, m \geq 0$.

We begin at level 1. At step 1 we initialize the auxiliary functions ag_1 and ag_2 as well as the polynomial coefficients $a(0, 0, n; x)$. At step 2 we compute the coefficients $a(1, 0, n; x)$ and $a(0, 1, n; x)$. These results may be represented in Figure 4.1.

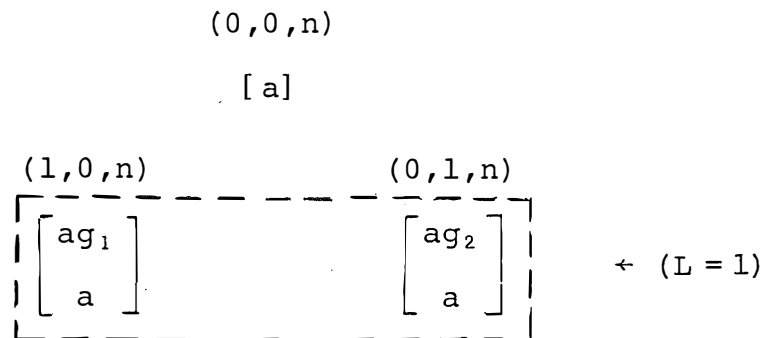


Figure 4.1

If $L = N$ the process terminates at step 3. Otherwise we begin the next level at step 4 by initializing ag_2 and ag_1 for the "boundary points" that were not computed at the previous level. The level counter L is incremented and "initial" auxiliary functions are computed for the next "level" of points. At step 5 we compute additional auxiliary functions needed to recursively compute the "initial" values on the edge of the triangular array (compare figure 3.1). The auxiliary functions for the next level of coefficients are computed in stage 6 and the coefficients themselves are computed in step 7. The values computed for $L = 2, 3$ are represented in Figure 4.2, 4.3 respectively.

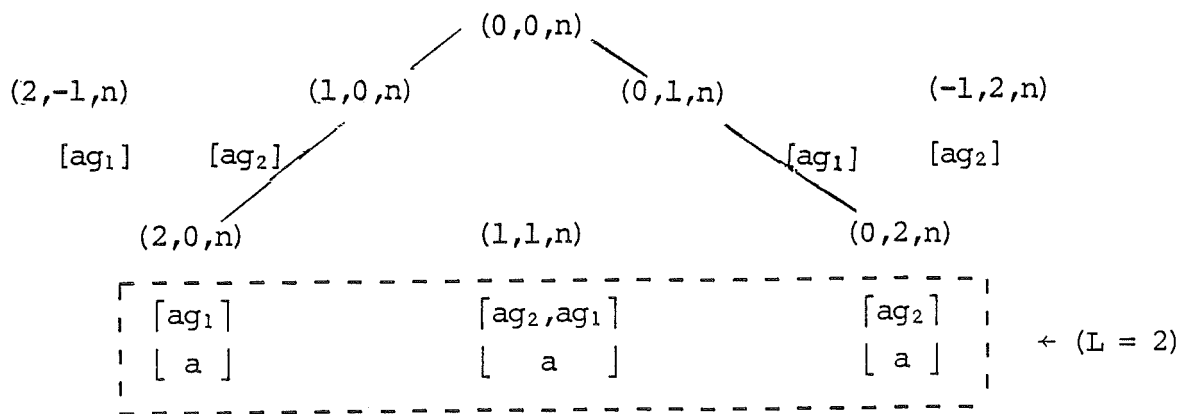


Figure 4.2

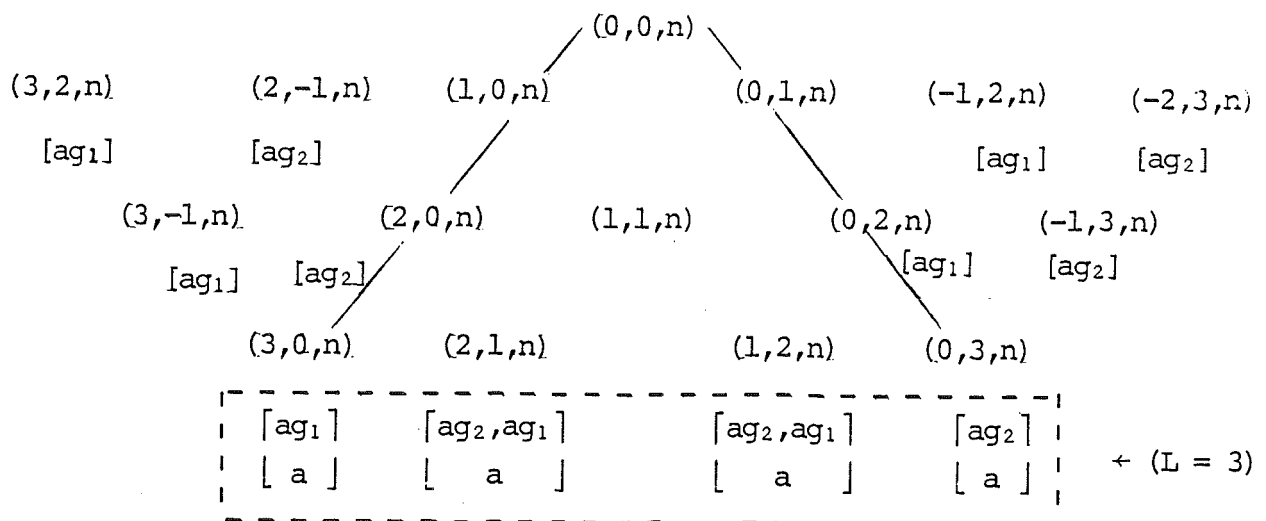


Figure 4.3

5. EXAMPLES

Example 1: Quadratic approximations to $f(x) = e^x$

In this example $N = 4$. The table gives the polynomial coefficients of the quadratic approximations of all orders up to 4. These are arranged by "levels" as calculated by the algorithm.

In addition the approximations have been evaluated at $x = 1$ to give an approximation to the value of e . The first or second root of the quadratic was chosen according as which was the closer root. Refer to table 5.1.

Table 5.1

(x_1 and x_2 refer to the 2 roots of the quadratic)

L	m	n	p	q	r	Value at x = 1		Exact value (2.7182818)
						X1	X2	
0	0	0	1	-2	1	1		
		1	1	-4	3+2x	-		
		2	1	-8	7+6x+2x ²	3		
		3	1	-16	15+14x+6x ² + $\frac{4}{3}$ x ³	2.7400887		
		4	1	-32	31+30x+14x ² +4x ³ + $\frac{2}{3}$ x ⁴	2.7209438		
1	0	0	1 - $\frac{2}{3}$ x	- $\frac{4}{3}$	$\frac{1}{3}$			3.7320508
		1	1 - $\frac{1}{2}$ x	-2	1 + $\frac{1}{2}$ x			3
		2	1 - $\frac{2}{5}$ x	- $\frac{16}{5}$	$\frac{11}{5} + \frac{8}{5}x + \frac{2}{5}x^2$			3
		3	1 - $\frac{1}{3}$ x	- $\frac{16}{3}$	$\frac{13}{3} + \frac{11}{3}x + \frac{4}{3}x^2 + \frac{2}{9}x^3$	2.7090056		
0	1	0	1	-2x	-1			2.4142136
		1	1	4-4x	-5-2x			2.6457513
		2	1	16-8x	-17-10x-2x ²			2.7082039
		3	1	48-16x	-49-34x-10x ² - $\frac{4}{3}$ x ³			2.7171935
2	0	0	1 - $\frac{6}{7}$ x + $\frac{2}{7}$ x ²	- $\frac{8}{7}$	$\frac{1}{7}$			2.5351838
		1	1 - $\frac{8}{11}$ x + $\frac{2}{11}$ x ²	- $\frac{16}{11}$	$\frac{5}{11} + \frac{2}{11}x$			2.6770320
		2	1 - $\frac{5}{8}$ x + $\frac{1}{8}$ x ²	-2	1 + $\frac{5}{8}x + \frac{1}{8}x^2$			2.7071068
1	1	0	1 - $\frac{2}{5}$ x	- $\frac{4}{5} - \frac{4}{5}x$	- $\frac{1}{5}$			2.7862996
		1	1 - $\frac{1}{3}$ x	- $\frac{4}{3}x$	-1 - $\frac{1}{3}x$			2.7320507
		2	1 - $\frac{2}{7}$ x	- $\frac{16}{7} + \frac{16}{7}x$	- $\frac{23}{7} - \frac{12}{7}x - \frac{2}{7}x^2$			2.7202941
0	2	0	1	-2-x ²	1			2.6180340
		1	1	-8+4x-2x ²	7+2x	3		3
		2	1	-32+16x-4x ²	31+14x+2x ²	2.7198901		
3	0	0	1 - $\frac{14}{15}x + \frac{2}{5}x^2 - \frac{4}{45}x^3$	- $\frac{16}{15}$	$\frac{1}{15}$			2.7843137
		1	1 - $\frac{11}{13}x + \frac{4}{13}x^2 - \frac{6}{117}x^3$	- $\frac{16}{13}$	$\frac{3}{13} + \frac{1}{13}x$			2.7843137
2	1	0	1 - $\frac{10}{17}x + \frac{2}{17}x^2$	- $\frac{16}{17} - \frac{8}{17}x$	- $\frac{1}{17}$			2.7077418
		1	1 - $\frac{12}{23}x + \frac{2}{23}x^2$	- $\frac{16}{23} - \frac{16}{23}x$	- $\frac{7}{23} - \frac{2}{23}x$			2.7164006
1	2	0	1 - $\frac{2}{7}x$	- $\frac{8}{7} - \frac{4}{7}x - \frac{2}{7}x^2$	$\frac{1}{7}$			2.7266500
		1	1 - $\frac{1}{4}x$	-2x - $\frac{1}{2}x^2$	1 + $\frac{1}{4}x$			2.7032574
0	3	0	1	-2x - $\frac{1}{3}x^3$	-1			2.7032574
		1	1	8-8x+2x ² - $\frac{2}{3}x^3$	-9-2x			2.7162972
4	0	0	1 - $\frac{30}{31}x + \frac{14}{31}x^2 - \frac{4}{31}x^3 + \frac{2}{93}x^4$	$\frac{32}{31}$	$\frac{1}{31}$	2.7112428		
3	1	0	1 - $\frac{409}{833}x + \frac{349}{1666}x^2 - \frac{17}{588}x^3$	- $\frac{818}{833} - \frac{529}{1666}x$	- $\frac{15}{833}$			2.7206005
2	2	0	1 - $\frac{14}{31}x + \frac{2}{31}x^2$	- $\frac{32}{31} - \frac{16}{31}x - \frac{4}{31}x^2$	$\frac{1}{31}$			2.7174743
1	3	0	1 - $\frac{2}{9}x$	- $\frac{8}{9} - \frac{8}{9}x - \frac{2}{9}x^2 - \frac{2}{27}x^3$	- $\frac{1}{9}$			2.7192031
0	4	0	1	-2-x ² - $\frac{1}{12}x^4$	1			2.7150107

Example 2: Quadratic approximations to

$$f(x) = [(1+2x)/(1+x)]^{\frac{1}{2}}$$

Since this function satisfies the equation

$$(1+x)f^2(x) - (1+2x) = 0$$

it is a quadratic function and should be able to be represented exactly by a (1,0,1) "approximations". Assuming the representations:

$$f(x) = 1 + \frac{1}{2}x - \frac{5}{8}x^2 + \frac{13}{16}x^3 - \dots$$

and $f^2(x) = g(x) = 1 + x - x^2 + x^3 - \dots$

with $N = 2$, the following table (table 5.2) of polynomial coefficients is obtained. In this case we stop at $L = 1$ since this gives the expected result, although there are other representations of this power series as a quadratic function.

L = 1	l m n			p	q	r
	0	0	0	-1	2	-1
	0	0	1	-1	$\frac{8}{5}$	$-\frac{3}{5} + \frac{1}{5}x$
	0	0	2	-1	$\frac{16}{13}$	$-\frac{3}{13} + \frac{5}{13}x - \frac{3}{13}x^2$
	1	0	0	$1 - \frac{1}{9}x$	$-\frac{16}{9}$	$\frac{7}{9}$
	1	0	1	$1 + x$	0	$-1 - 2x$
	0	1	0	1	$-\frac{12}{7} - \frac{1}{7}x$	$\frac{5}{7}$
	0	1	1	1	$8 + 12x$	$-9 - 17x$

Table 5.2

6. CONCLUSION

This paper considers the extension of the Pade approximation to a higher dimension, motivated by the work of Shafer [13]. This increase in dimension inevitably leads to a more complicated computation. The primary goal was to construct a recursive computational scheme for these quadratic approximations.

A similar basic idea has been previously studied [4]. The authors compute the coefficients of the polynomial coefficients of the more general Padé - Hermite approximation. This computation involved determinants of order $(l+m+n+3)$.

More recently, a more extensive discussion of the Padé-Hermite approximations along similar lines has been given by Paszkowski [11,12].

In this paper the polynomial coefficients are expressed by determinants of order $(l+m+2)$. The computation is analogous to the Neville-Aitken scheme for computing the polynomial approximation determined by collocation. The general algorithm is based on Brezinski's algorithm [2],[3].

The generalization of the Pade approximation to quadratic approximation leads to a generalization of the epsilon algorithm, and hence to new methods of accelerating the convergence of slowly convergent series. This application will be the subject of a subsequent paper.

REFERENCES

1. G. A. BAKER, *Essentials of Pade Approximants*, Academic Press, (1975).
2. C. BREZINSKI, *The Muhlbach-Neville-Aitken algorithm and some extensions*, BIT 20 (1980), pp.444-451.
3. C. BREZINSKI, *A general extrapolation algorithm*, Numer. Math. 35 (1980), pp. 175-187.
4. J. DELLA DORA & C. DI-CRESCENZO, *Approximation De Pade-Hermite, in Pade Approximation and its Applications*. L. Wuytack ed., Lecture Notes in Mathematics 765, Springer-Verlag, Heidelberg, 1979, pp. 88-115.
5. W. B. GRAGG, *The Pade table and its relation to certain algorithms of Numerical Analysis*, SIAM Rev. 14, (1972), pp. 1-62.
6. JEZIORSKI, SCHWALM AND SZALEWICZ, *Analytic continuation in exchange perturbation theory*. J. Chem. Phys. 73 (1980), pp. 6215-6224.
7. S. L. LOI AND A. W. MCINNES, *An algorithm for generalised rational interpolation*, BIT 23 (1983), pp. 105-117.
8. S. L. LOI, *A general algorithm for Pade approximation to formal power series*. Math research report No. 22, University of Canterbury, 1982.
9. S. L. LOI, *Quadratic approximation and its application to acceleration of convergence*. Ph.D. thesis, University of Canterbury, 1982, pp. 132-188.
10. G. MUHLBACH, *The general Neville-Aitken algorithm and some applications*, Numer. Math. 31 (1978), pp. 97-110.
11. S. PASZKOWSKI, *Quelques algorithmes de l'approximation de Padé-Hermite*, Publication ANO-89, Université de Lille, 1983.
12. S. PASZKOWSKI, *Quelques generalisations de l'approximation de Padé*, Publication ANO-93, Université de Lille, 1983.

13. R. E. SHAFER, *On quadratic approximation*, SIAM J.Num.Anal. 11 (1974), pp. 447-460.
14. S.L. SHORT, *The evaluation of Feynman integrals in the physical region using multi-valued approximants*. J. Phys. G-Nu 5 (1979), pp. 167-198.